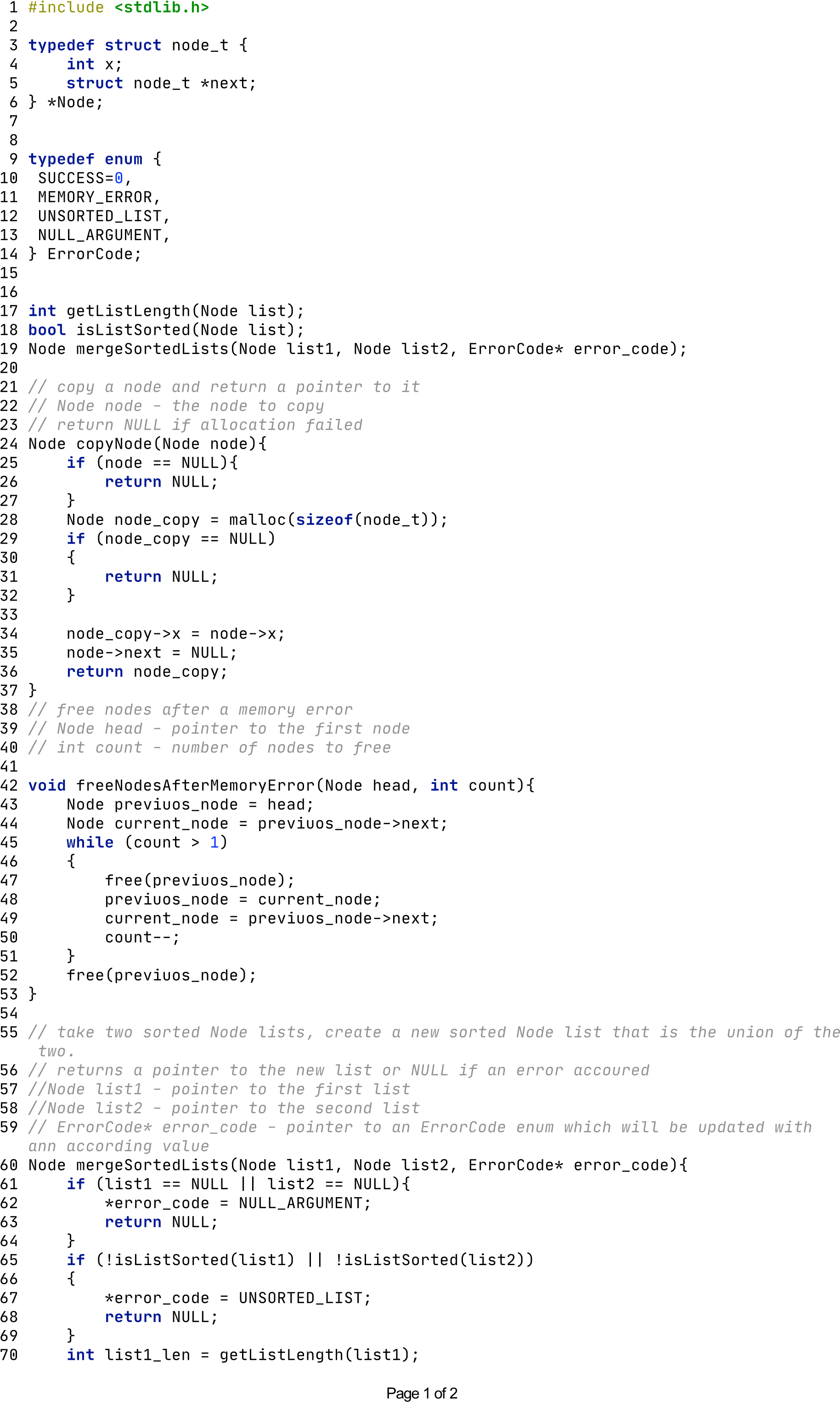
Dry – Part 1





Dry – part 2

**מצורף תיקון אח לכל התרגיל לאחר פירוט השגיאות☺**

**שגיאות:**

1. char\* str2;

המצביע לא אותחל לכן יש בו ערך זבל בשלב זה.

זו שגיאת זמן ריצה במקרה הטוב והתנהגות בלתי מוגדרת במקרה הרע.

ניתן לתקן ע"י אתחול ל-NULL וככה נקבל שגיאת זמן ריצה, במקום התנהגות בלתי מוגדרת, כשננסה לגשת לכתובת 0. אבל תמיד נעדיף לאתחל לכתובת כלשהי, שלא נשכח בהמשך.

1. x = strlen(str);

x- מטיפוס int\*

strlen(str) – מטיפוס int

אין המרה אוטומטית כזאת בשפת C, לכן, שגיאת קומפלציה.

1. str2 = malloc(\*x);

שגיאת זמן ריצה כי לא הוקצה מקום במאלוק עבור התו '\'

1. בנוסף, אין בדיקה של הערך המוחזר ממאלוק אחרי שגיאה מס' 3. כלומר, כאשר בלולאת ה-for מנסים לגשת לכתובת str2 בפקודה str2[i] = str[\*x - i];, אם מאלוק נכשל אז ניגשים לכתובת לא חוקית. שגיאת זמן ריצה.
2. הלולאה for אינה בסוגריים מסולסלים. לפי קונבנציות הקורס, צריכה להיות גם אם זו פקודה אחת- חריגה מכללי תכנות נכון.
3. שם הפונק' foo, אינו משמעותי ואינו מעיד על פעולת הפונק'- חריגה מכללי תכנות נכון. לפי קונבנציות הקורס, שם הפונק' צריך להתחיל בפועל ושיעיד על מה שהפונק' עושה. כנ"ל לגבי המצביע X שעליו להיות שם משמעותי.
4. str2[i] = str[\*x - i]; יש חריגה מגבול מערך התווים str כבר בריצה ראשונה, כי נניח אורך str שווה ל-3, אזי בריצה ראשונה של הלולאה נקבל ש- str2[0] = str[3]; והאינדקס האחרון ב-str הוא 2. שגיאת זמן ריצה.
5. בשורות ההדפסה אין ירידת שורה חדשה, וזה פחות טוב כי אם יש למשל קריאה לפונקציה foo מתוך פונק' אחרת שגם היא מדפיסה אחרי ההדפסה בפונק' foo אז זה יהיה צמוד לאותה שורה שהודפסה בפונק' הנ"ל. חריגה מכללי תכנות נכון.
6. בנוסף, סדר ההדפסות הפוך ממה שהפונק' אמורה לעשות. התנהגות בלתי מוגדרת.

**תיקון:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//added const to char\* str to be clear that the str is not supposed change

char\* reverseStringAndPrint(const char\* str, int\* ptr\_length){

if(!str || ! ptr\_length){

return NULL;

}

char\* reversed\_str= malloc(strlen(str)+1);

if(!reversed\_str){

return NULL;

}

\*ptr\_length =strlen(str);

int length= \*ptr\_length; //easier to deal with basic parameter than ptr

for(int i=0; i< length; i++){

reversed\_str[i] = str[length-i-1];

}

if(length %2==0){

printf(“%s/n”, reversed \_str);

}

else{

printf(“%s/n”, str);

}

return reversed\_str;

}